

# A combination of STDP, Hebbian learning and synaptic scaling deriving from a theoretical learning principle.

Mathieu N. Galtier<sup>1</sup>, Gilles Wainrib<sup>2</sup>

**1 School of Engineering and Science, Jacobs University Bremen gGmbH, P.O. Box 750 561, 28725 Bremen, Germany**

**2 Laboratoire Analyse Géométrie et Applications, Université Paris 13, 99 avenue Jean-Baptiste Clément, Villetaneuse, France**

## Abstract

We introduce a general learning principle designed to capture the dynamics of an input into a recurrent neural network. By minimizing the relative entropy between a given input dynamical system and the spontaneous activity of the neural network, we derive a new unsupervised learning rule. This learning rule can be interpreted in terms of biologically relevant synaptic plasticity mechanisms, such as spike-timing dependent plasticity and synaptic scaling.

## Author Summary

Learning in the brain is generally thought to be mediated by local changes in the neuronal network connectivity occurring at the synapses. The connections between neurons strengthen or weaken according to learning rules exclusively based on the activity of the neurons they are connected with. In this paper, we introduce a learning rule which gives an explicit way for the connections to tune themselves so that the global network learns to copy and reproduce the dynamics of its stimuli. Indeed, it had been advocated that the spontaneous activity of a biological neural networks may replay the stimuli the network was previously exposed to. Based on this hypothesis, we apply several mathematical tools on a simplified network model to identify an optimal learning rule. Although the approach is mainly theoretical, we show how the proposed learning rule can be interpreted in terms of biologically plausible mechanisms. This suggests a concrete method for the brain to build itself a model of the world.

## Introduction

For several decades, understanding the synaptic mechanisms underlying learning processes in the brain has been an active field of research, starting from the pioneering work of Hebb [1] to the latest discoveries on spike-timing dependent plasticity (STDP) initiated by [2, 3]. Learning is mainly thought to rely on the modifications of synapses strength caused by co-activations of connected neurons, reflecting in particular correlations and causalities in the inputs structure. Beyond these modifications rules, learning is also entwined with key homeostatic plasticity processes regulating the overall level of neuronal activity [4, 5].

In terms of modeling, two main classes of learning rules can be distinguished: (i) those based on biological observations with an aim to clarify their functional purposes, e.g. the BCM learning rule [6], or the STDP with a recent interpretation proposed in [7] and (ii) those designed from theoretical principles whose link with biology is determined a posteriori [8, 9, 10]. In the second class of learning rules, the theoretical principle often corresponds to optimize a certain quantity [11]. The choice of this quantity is crucial and leads to qualitatively different results.

Recent experimental findings [12, 13] have shown that spontaneous activity patterns in the brain were very similar to the evoked activity. It suggests that the spontaneous activity collects dynamical statistics about its environment and may even replay the stimuli it was exposed to. Although biological experiments usually compare spontaneous and evoked activity, it also makes sense from a theoretical perspective to compare the spontaneous dynamics of the network directly to the dynamics of its inputs.

Therefore, we propose to derive a learning rule which minimizes a specific distance between the autonomous activity of a rate-based neural network and another dynamical system which has produced the inputs to the network during the learning phase. The distance we choose between the two dynamical systems is based on an analogy to the relative entropy or Kullback-Leibler divergence. In the theory of stochastic processes, the relative entropy between two diffusion processes appears as a natural way to estimate a discrepancy between their laws in the path space. This approach is closely related to the free energy principle [14] since it corresponds to minimizing the surprise of the network to its sensory inputs.

Minimizing this distance may lead to very accurate approximations of the inputs, because it has been proved that any dynamical system can be arbitrarily finely approximated by such a network, provided there is a sufficient number of neurons [15]. Besides, the minimization of the Kullback-Leibler divergence between two stationary Gibbs measures, has proven very fruitful in the context of Boltzmann machines [16, 17] leading to state of the art methods in data mining. In contrast with Boltzmann machines, our choice is tailored to capture dynamical properties of the inputs, rather than static statistical features.

In this article, we intend to derive a biologically relevant learning rule from this theoretical principle. First, we express the relative entropy between the input and the network activity as a function of the connectivity matrix. Computing the gradient of this quantity with respect to the connectivity leads to a gradient descent algorithm to update the connectivity. We then introduce a learning rule approximating the gradient descent in the slow-fast asymptotic regime. We also illustrate numerically the properties of this learning rule on three examples. Finally, we discuss the biological relevance of the learning rule and future extensions.

## Results

We consider a neural network made of  $n$  neurons which is exposed to a time dependent input  $\mathbf{u}(t) \in \mathbb{R}^n$  of the same dimension. As our approach is focused on learning the dynamics of the input, we assume that  $\mathbf{u}$  is generated by an arbitrary dynamical system:

$$\dot{\mathbf{u}} = \xi(\mathbf{u}) \quad (1)$$

with  $\xi$  a smooth vector field from  $\mathbb{R}^n$  to  $\mathbb{R}^n$ . We also assume that the trajectory of the inputs is  $\tau$ -periodic. Although the key mathematical assumptions on the input  $\mathbf{u}$  are stationarity and ergodicity, we restrict our study to periodic inputs for simplicity. We comment on possible extensions to a stochastic setting in the **Discussion**.

Although the following method virtually works with any network equations (see **Discussion**), we focus on a neural network composed of  $n$  neurons and governed by

$$\dot{\mathbf{v}} = -l\mathbf{v} + \mathbf{W}.S(\mathbf{v}) + \mathbf{u}(t) \quad (2)$$

where  $\mathbf{v} \in \mathbb{R}^n$  is a vector representing neuronal activity,  $\mathbf{W} \in \mathbb{R}^{n \times n}$  is the connectivity matrix,  $l \in \mathbb{R}_+$  is a decay constant and  $S$  is an entry-wise sigmoid function. The spontaneous activity of the network is given by the same equation without the term  $\mathbf{u}(t)$ . This setup is illustrated in Figure 1.

## Learning rule derivation

Inspired from the definition of relative entropy between stochastic processes (see **Material and methods**) we define

$$H_{\mathbf{u}} := \frac{1}{2} \int_0^\tau \left\| -l\mathbf{u}(s) + \mathbf{W}.S(\mathbf{u}(s)) - \xi(\mathbf{u}(s)) \right\|^2 ds \quad (3)$$

When  $H_{\mathbf{u}} = 0$ , the vector fields of systems (1) and (2) are equal on the trajectories of the inputs. This quantity may be viewed as a distance between the two vector fields defining the dynamics of the inputs and of the neuronal network along the trajectories of the inputs. Note that it is similar to classical gradient methods [11], except that the norms are applied to the flows of inputs and neural network instead of their activity. Thus, it focuses more specifically on the dynamical structure of the inputs.

In order to capture the dynamics of the inputs into the network, it is natural to look for a learning rule minimizing this quantity. To this end, we consider the gradient of this measure with respect to the connectivity matrix:

$$\nabla_{\mathbf{W}} H_{\mathbf{u}} = - \left[ \xi(\mathbf{u}) \cdot S(\mathbf{u})' + l\mathbf{u} \cdot S(\mathbf{u})' - \mathbf{W} \cdot S(\mathbf{u}) \cdot S(\mathbf{u})' \right] \quad (4)$$

where  $\{\mathbf{x} \cdot \mathbf{y}'\}_{ij} = \int_0^\tau \mathbf{x}_i(s) \mathbf{y}_j(s) ds$  for  $\mathbf{x}, \mathbf{y}$  functions from  $[0, \tau]$  to  $\mathbb{R}^n$ . Thus, an algorithm implementing the gradient descent  $\dot{\mathbf{W}} = -\nabla_{\mathbf{W}} H_{\mathbf{u}}$  is a good candidate to minimize the relative entropy between inputs and spontaneous activity. One can easily show that  $\mathbf{W} \rightarrow H_{\mathbf{u}}$  is a convex function, thus excluding situations with multiple local minima. Moreover, if  $S(\mathbf{u}) \cdot S(\mathbf{u})'$  is invertible, one can compute directly the minimizing connectivity as  $\mathbf{W}^* = [\xi(\mathbf{u}) \cdot S(\mathbf{u})' + l\mathbf{u} \cdot S(\mathbf{u})'] \cdot [S(\mathbf{u}) \cdot S(\mathbf{u})']^{-1}$ .

In terms of biological relevance, there are three main issues linked with equation (4). First, it is a batch learning algorithm which requires an access to the entire history of the inputs. Second, it requires a direct access to the inputs  $\mathbf{u}$ , whereas synaptic plasticity mechanisms shall only rely on the network activity  $\mathbf{v}$ . Third, it requires the ability to compute  $\xi(\mathbf{u})$ , the time-derivative of the inputs.

Thus, we want to design a new learning rule mimicking equation (4) which would avoid these problems. First, this new learning rule is online, i.e. it takes input on the fly, and relies on a slow/fast mechanism to temporally average the variables. Second, it uses an estimate of the inputs, noted  $\bar{\mathbf{v}}$ , which is based on the activity variable only. Third, it approximates the computation of  $\xi(\mathbf{u})$  with a function  $\delta$  inspired from STDP convolution. This learning rule can be written

$$\frac{1}{\epsilon} \dot{\mathbf{W}}_{ij} = \delta[\bar{\mathbf{v}}_i, S(\bar{\mathbf{v}}_j)] + l\bar{\mathbf{v}}_i S(\bar{\mathbf{v}}_j) - \sum_k \mathbf{W}_{ik} S(\bar{\mathbf{v}}_k) S(\bar{\mathbf{v}}_j) \quad (5)$$

where  $\epsilon \ll 1$  is the learning rate,  $\delta[x, y] = \frac{\gamma}{2} (x(y * g_\gamma) - (x * g_\gamma)y)$  and  $\bar{\mathbf{v}} = l\mathbf{v} - \mathbf{W} \cdot S(\mathbf{v}) * g_l$ . The function  $g_c$  is defined as  $g_c : t \mapsto ce^{-ct} H(t)$  where  $c$  is a positive number. The constant  $\gamma \in \mathbb{R}_+$  is a time constant corresponding to the width of the STDP window used for learning. A schematic representation of (5) is displayed in Figure 2. We give a full justification of the link between this rule and (4) in **Materials and methods**. The online equation (5) is a good approximation of the gradient descent (4) if the following assumption is verified

$$\mathbf{u} \simeq \mathbf{u} * g_l * g_\gamma \quad (6)$$

i.e. the inputs are slow compared to the time scales of the network. If this assumption is not verified then the network only learns to replay the slow variations of the inputs.

A striking feature of this new learning rule is that the three terms appearing in (5) have a biological interpretation in terms of STDP, Hebbian learning and synaptic scaling, as explained in details the **Discussion**.

## Applications

We now present three examples which illustrate the above learning rule.

### Proof of concept

In order to illustrate the idea that learning rule (4) enables the network to learn dynamical features of the input, we have constructed the following experiment. We present to the network an input movie

displaying sequentially the writing of the letter A (figure 3.a - top row). To each pixel we assign one neuron, so that the input and the network share the same dimension. This input movie is repeated periodically until the connectivity matrix of the network, evolving under rule (4), stabilizes (figure 3.b). Then the input is turned off and we set the initial state of the network to a priming image showing the bottom left part of letter A. The network evolving without input strikingly reproduces the dynamical writing of letter A as displayed in figure 3.a - bottom row. A preliminary investigation of the eigenvalues and eigenvectors of the final connectivity matrix reveals that the number of relevant modes (non-zero eigenvalue) is slightly below the number of time frames used in the input movie, indicating an efficient information storage into the connectivity. Thus, with this example we have illustrated the ability of the learning rule we have derived from a theoretical principle to capture a dynamic input into a connectivity matrix.

### From activity to connectivity

We now investigate the question of retrieving the connectivity of a neural network based on the observation of the time series of its activity. This is an inverse problem which is one of the most challenging topics in computational neuroscience [18, 19, 20] since it may give access to large scale effective connectivities simply from the observation of a neuronal activity. Here we address it in an elementary framework. The network generating the activity patterns is referred as input network and evolves according to  $\dot{\mathbf{u}} = -l_0 \mathbf{u} + \mathbf{W}_0.S(\mathbf{u})$ . For this example, the network is made of  $n = 3$  neurons and its connectivity  $\mathbf{W}_0$  is shown in figure 4.a). These parameters were chosen so that the activity is periodic as shown by the dashed curves in figure 4.c). Then, we simulate both equations (2) and (5) with a decay constant  $l_{net}$  and observe that its connectivity  $W_{net}$  converges to  $\mathbf{W}_0$ .

As opposed to the batch learning rule (4), it is necessary for assumption (6) to be satisfied in order to approximate accurately the input's activity with the online learning rule (5). Given that the time constant of the inputs is governed by  $l_0$ , the previous approximation holds only if  $\gamma, l_{net} \gg l_0$ . If this assumption is broken, e.g.  $l_{net} = l_0$ , then the final connectivity matrix is different from  $\mathbf{W}_0$ , see the red dot-dashed curve in figure 4.b). Indeed, in this case, the network only learns to replay the slow variation of the inputs.

A method to recover the precise time course of the inputs consists in artificially changing the time constants at different steps of an algorithm described in the following. First, simulate the network equation (2) with a constant  $l_{net} \gg l_0$ . Yet, the time constant in the learning equation (5) is to be kept at  $l_0$ , thus introducing a hybrid model. In this framework, the connectivity converges exactly to  $\mathbf{W}_0$  as shown in the blue dashed curve in figure 4.b). After learning, simulate the network equation (2) with the learned connectivity and with a time constant switched back to  $l_0$ . This gives an activity as shown in the plain curves in figure 4.c).

### Learning the dynamics

The last example extends the approach by devoting an arbitrarily large number of neurons to the processing in contrast to the two previous examples. Indeed, the computational power of neural networks of the type (2) lies in the possibility of having a huge amount of neurons. However, the learning rule (5) implicitly assumes that the network and its inputs have the same dimensionality. Indeed, if a neuron receives no input then a close inspection of (15) shows that, at equilibrium, it does not receive connections from other neurons. Therefore, learning low dimensional, yet temporally rich, inputs leads to poor results. To overcome this technical issue, low dimensional inputs need to be projected in higher dimensions before they can be accurately learned by the recurrent neural network introduced above. An example of such a projection, corresponds to considering that neurons code for the position of the input in its phase space. For instance, a neuron might receive excitation only when the input is in a small ball centered around the origin. This projection somewhat corresponds to the discretization of the transport equation equa-

tion derived from (1). The left picture of figure 5.b illustrates this projection for the following example. Assume that the input trajectory is given by a point  $(x(t), y(t)) = (-3 \cos(t), \sin(t) + \cos(2t))$  as shown by the blue curves in the top picture of figure 5.a. This particular input is chosen because its trajectory in the phase space does not intersect itself and thus it is the solution of a dynamical system (which we do not need to explicit). Besides, if we try to approximate this two dimensional input with a network of two neurons, then the spontaneous activity post-learning is a very bad approximation of the signal, as shown by the red curves in the middle picture of figure 5.a. Indeed, the two-neurons networks of type `eqrefeq`: voltage based do not span the dynamics provided by the inputs (which has different mode of oscillations). However, we show below that increasing the number of neurons leads to a much better approximation as shown in the bottom picture of figure 5.a.

In the high dimensional space it is possible to compare the dynamics of the projected inputs  $p(\mathbf{u})$  and the dynamics  $\mathbf{v}$  as  $d(\mathbf{u}, \mathbf{v}) = \frac{1}{n_{eff}T} \int_0^{2\pi} \|p(\mathbf{u}(s)) - \mathbf{v}(s)\|_1 ds = \frac{1}{n_{eff}T} \sum_{s=0}^T \|p(\mathbf{u}(\frac{2\pi s}{T})) - \mathbf{v}(\frac{2\pi s}{T})\|_1$ , where  $n_{eff}$  is the number of neurons which receive a non-null input and  $T = 50$  is the number of time steps used for discretization. In the right picture of figure 5.b, we show that the accuracy decreases to zero with the number of neurons in the network. When the error is null we see on the red curve that the number of non-quiet neurons is always equal to  $T$ , which means that every point of the inputs has been “boxed” by separate neurons.

The equilibrium connectivities for different number of neurons are represented in figure 5.c. We observe that when the number of neurons is large enough for the representation of the input to be optimal (at machine precision), i.e. the right picture corresponding to  $n = 3364$  neurons, the neurons are strongly self inhibitory and send their excitation to one neuron. Actually, they pass their excitation to the next neuron and become immediately quiet due to self inhibition. When there are less neurons in the network, i.e. the left picture corresponding to  $n = 100$  neurons, the behavior of the network is similar except that the self inhibition of some neurons is not sufficient to suppress their excitation in one time step. Therefore, they will be excited during a longer time. This corresponds to the fact that two different time steps of the inputs lead to exciting the same neuron. Therefore, the accuracy is poorer, see right picture of figure 5.b.

The projection we introduced is a classical precomputation trick used in many algorithms. We suggest that it may correspond to the processing performed by vertical connections connecting the senses to the cortex, e.g. retino-cortical connections. Indeed, in the primary visual cortex it is well-known that some neurons code for local oriented edges in the visual field. The receptive fields of the neurons cover all the space position plus orientation.

## Discussion

### Biological interpretation

The learning rule (5) introduced in this article can be interpreted from a biological point of view. We now discuss the different terms in (5) and relate them to the biologically oriented litterature.

The variable  $\bar{\mathbf{v}}$  can be seen as a spatio-temporal differential variable. Although unsupervised learning rules are often algebraic combinations of element-wise functions applied to the activity of the network [21], it is not precisely the case here. Indeed, learning is based on the variable  $\bar{\mathbf{v}}$  which corresponds to the subtraction of the temporally integrated synaptic drive  $\mathbf{W}.S(\mathbf{v}) * g_l$  from the activity of the neurons  $l\mathbf{v}$ . For each neuron, this variable takes into account the past of all the neurons which are then spatially averaged through the connectivity to be subtracted from the current activity. This gives a differential flavor to this variable which is reminiscent of former learning rules [6, 22] for the temporal aspect and [23] for the spatial aspect. Note that this variable is not strictly speaking local (i.e. the connection  $\mathbf{W}_{ij}$  needs the values of  $\mathbf{v}_k$  to be updated), yet it is biologically plausible since the term  $(\mathbf{W}.S(\mathbf{v}))_i$  is accessible for neuron  $i$  on its dendritic tree, which is a form of locality in the broader sense.

The first term  $\delta[\bar{\mathbf{v}}_i, S(\bar{\mathbf{v}}_j)]$  in (5) can be related to antisymmetric STDP learning [24]. This term is antisymmetric and is responsible for retrieving the drift  $\xi$  in the mathematical computations above. Thus it captures the causality structure of the inputs which is a task generally attributed to STDP [25]. Beyond the simple similarity of functional role, we believe a simplification of this term may shed light on the deep link it has with STDP. The main difference between our setup and STDP is that the former is based on a rate-based dynamics (2), whereas the latter is based on a spiking dynamics. In a pure spike framework, i.e the activity is a sum of Diracs, the STDP can be seen as this simple learning rule  $\dot{\mathbf{W}}_{ij} \propto \delta[\mathbf{v}_i, \mathbf{v}_j] \propto \mathbf{v}_i(\mathbf{v}_j * g_\gamma) - (\mathbf{v}_i * g_\gamma)\mathbf{v}_j$ . Indeed, the term  $\mathbf{v}_i(\mathbf{v}_j * g_\gamma)$  is non-null only when the post-synaptic neuron  $i$  is firing and then, via the factor  $\mathbf{v}_j * g_\gamma$ , it counts the number of preceding pre-synaptic spikes that might have caused  $i$ 's excitation and weight them by the decreasing exponential  $g_\gamma$ . Thus this term exactly account for the positive part of the STDP curve. The negative term  $-(\mathbf{v}_i * g_\gamma)\mathbf{v}_j$  takes the opposite perspective and accounts for the negative part of the STDP curve. A loose extension of this rule to the case where the activity is smoothly evolving leads to identifying the function  $\delta$  to the STDP mechanism for rate-based networks [26, 27].

The second term  $l\bar{\mathbf{v}}_i S(\bar{\mathbf{v}}_j)$  in (5) is a form of Hebbian learning. This term is similar to the original Hebbian learning rule [1, 28, 29]. It corresponds to the co-activation of the pre- and post-synaptic cells. Actually, it can also be seen as the symmetric part of the STDP temporal profile complementing the purely antisymmetric STDP of the previous term. This is better seen if one slightly changes the learning rule (5) by replacing this term by  $l(\bar{\mathbf{v}}_i * g_\gamma)(S(\bar{\mathbf{v}}_j) * g_\gamma)$  which would not change the qualitative behavior of the learning rule, e.g. it would still minimize the relative entropy. In this case and based on the **Material and methods**, it is easy to combine it with the  $\delta$  function such that it becomes  $\delta[x, y] = \frac{\gamma+l}{2}x(y * g_\gamma) - \frac{\gamma-l}{2}(x * g_\gamma)y$ , which corresponds to an asymmetric STDP learning rule [24].

The third term  $-\sum_k \mathbf{W}_{ik} S(\bar{\mathbf{v}}_k) S(\bar{\mathbf{v}}_j)$  in (5) accounts for what is usually presented as homeostatic plasticity mechanisms. Although the two previous terms (STDP and Hebbian learning) seem to be powerful mechanisms to shape the response of the network, there is a need of a regulatory process to prevent from uncontrolled growth of the network connectivity [4, 5, 30]. It has been argued that STDP could be self regulatory [31, 32], but it is not the case in our framework and an explicit balancing mechanism is necessary to avoid the divergence of the system. This last term is the only one with a negative sign and is multiplicative with respect to the connectivity. Thus, according to [4], it is a reasonable candidate for homeostasis. It has been argued [33, 34] that homeostatic plasticity might keep the relative synaptic weights by dividing the connectivity with a common scaling factor, theoretically preventing from a possible information loss. In contrast to these *ad hoc* re-normalizations often introduced in other learning rules [23, 35], our relative entropy minimizing learning rule thus introduces naturally an original form of homeostatic plasticity. Although we have separated the description of the three terms in (5), our approach suggests that homeostasis may be seen, not necessarily as a scaling term, but as a constitutive part of a learning principle, deeply entangled with the other learning mechanisms.

## Extension and perspectives

Although our approach eventually aims at addressing stochastic inputs, there are several conceptual and technical issues to extend our approach to noisy versions of (1) and (2). In the present article, we have introduced the quantity  $H_{\mathbf{u}}$  to measure the distance between the dynamics of the inputs and the spontaneous activity of the network after learning. This quantity is based on an analogy with the notion of relative entropy between the laws of two solutions of different stochastic differential equations sharing the same diffusion coefficient. Yet, the definition does not hold for different diffusion coefficients which is, a priori, our case. Thus, one may think that the network should learn both the vector field through  $\mathbf{W}$  and the diffusion matrix. Indeed, the network should learn also the covariant statistics of the input fluctuations to be able to reproduce similar patterns after the learning phase. An interesting perspective in this direction might be to implement a form of fluctuation-dissipation relationship to extract the relevant input statistics with an online learning rule. Although promising from an algorithmic viewpoint,

biological relevance of such a form of dual learning seems difficult to establish. Another technical point is that our construction of  $\mathbf{W}$  through (5) relies on the use of a STDP-inspired learning to estimate the time derivative  $\xi(\mathbf{u})$  of the inputs. However, if the inputs are perturbed by noise, then this estimation becomes less accurate.

Moreover, it appears that our general approach can be applied to other forms of network equations than (2) such as the Wilson-Cowan or Kuramoto models, leading to different learning rules. In this perspective, learning can be seen as a projection of a given arbitrary dynamical system to a versatile neuronal network model, and the learning rule will depend on the chosen model. However, we shall remark that any network equation with an additive structure – intrinsic dynamics + communication with other neurons – as in (2) will lead to a similar structure for the learning rule, with three terms that may share similar biological interpretations as developed above. A special case is the linear network  $\dot{\mathbf{v}} = -l\mathbf{v} + \mathbf{W}.\mathbf{v} + \mathbf{u}(t)$  for which various statistical methods to estimate the connectivity matrix have been applied e.g. in climate modeling [36], gene regulatory networks [37] and spontaneous neuronal activity [19]. The method developed in this article may be used to extend the *inverse modeling* approach previously developed in the linear case to models with non-linear interactions.

## Conclusion

In this article, we have identified a theoretical link between various highly debated concepts in neuroscience : Hebbian learning, spike-timing dependent plasticity and synaptic scaling. We have proved that an appropriate combination of these local mechanisms leads the network to learn to replay and predict its inputs. Not only this article focuses on the biological plausibility of the new learning rule, but it also provides a constructive way to build a network which reproduces the dynamics of any ergodic system.

## Materials and Methods

### Relative entropy

The quantity minimized through learning is defined in (3) and can be interpreted as the measure of the distance between the flows of the inputs and the neural network along the trajectories of the inputs. Although it is the starting point of our analysis, it is related and inspired by the relative entropy or the Kullback-Leibler divergence between two stochastic processes. These two stochastic processes are assumed to be the solution of a stochastic differential equation with the same diffusion term. This point explains why we could not apply directly the relative entropy minimization to our problem, since there is no reason why inputs and neural network should share the same diffusion term. Therefore, we restrict our study to deterministic systems and borrow the analogous of relative entropy to the field of stochastic processes.

We now introduce the computation of the relative entropy between two real diffusion processes  $x^f$  and  $x^g$  sharing the same diffusion coefficient. The extension to multidimensional stochastic processes is straightforward. Let  $\{\Omega, \mathcal{F}, P, \{\mathcal{F}_t\}_{0 \leq t \leq T}\}$  a probability space, equipped with the natural filtration of the standard Brownian motion. Consider a diffusion process  $(x_t)$  in  $\mathbb{R}$  under  $P$ , solution of the following stochastic differential equation (SDE):

$$dx_t = f(x_t)dt + \sigma dB_t \quad (7)$$

where  $B_t$  is a  $P$ -brownian motion. By Girsanov theorem, if  $Q$  is another probability absolutely continuous with respect to  $P$  and such that:

$$\frac{dQ}{dP} = \exp \left( \int_0^T \frac{g(x_s) - f(x_s)}{\sigma} dB_s - \frac{1}{2} \int_0^T \left[ \frac{g(x_s) - f(x_s)}{\sigma} \right]^2 ds \right) \quad (8)$$

then, under the technical condition that  $\int_0^T \left[ \frac{g(x_s) - f(x_s)}{\sigma} \right]^2 ds < \infty$   $P$ -a.s, one obtains a diffusion process under  $Q$ , solution of :

$$dx_t = g(x_t)dt + \sigma d\tilde{B}_t \quad (9)$$

where  $\tilde{B}_t$  is a  $Q$ -brownian motion. In other words, changing the drift in a diffusion process with a constant diffusion coefficient is equivalent to a change of the underlying probability. Notice that if  $f \equiv g$ , then  $\frac{dQ}{dP} = 1$  and  $P = Q$ .

We define then the relative entropy between the two diffusions

$$dx_t^f = f(x_t^f)dt + \sigma dB_t \quad (10)$$

and

$$dx_t^g = g(x_t^g)dt + \sigma dB_t \quad (11)$$

as the relative entropy between  $Q$  and  $P$ :

$$H(x^g|x^f) = H(Q|P) := \mathbb{E}_Q \left[ \ln \left( \frac{dP}{dQ} \right) \right] \quad (12)$$

An elementary computation shows that:

$$H(x^g|x^f) = \frac{1}{2} \mathbb{E} \left[ \int_0^T \left[ \frac{g(x_s^g) - f(x_s^g)}{\sigma} \right]^2 ds \right] \quad (13)$$

Similarly we can define  $H(x^f|x^g)$  by exchanging  $f$  and  $g$ , which gives a different quantity. Therefore the relative entropy between two stochastic processes is not a distance. Yet it gathers interesting information about the similarity between the two processes. A limitation of this definition is that it can only be defined in the case where both diffusion processes  $x^f$  and  $x^g$  share the same diffusion matrix  $\sigma$ . Otherwise the two associated measures  $P$  and  $Q$  are not absolutely continuous with respect to each other. However, this definition can interestingly be extended to the case of noise-free dynamical systems, which is the starting point of our definition of  $H_{\mathbf{u}}$  in equation (3).

## Justification of the online learning rule

The online learning rule (5) is expressed by means of the activity of the neural network  $\mathbf{v}$  governed by (2). However, to be comparable to the gradient of the relative entropy (4), we first need to make explicit the dependency on the inputs  $\mathbf{u}$ . Therefore, we show that the network dynamics induces a simple relation between  $\bar{\mathbf{v}}$  and the inputs  $\mathbf{u}$ : a simple computation in the Fourier domain shows that the convolution between the temporal operator  $\frac{d}{dt} + lI_d$  and  $g_l$  results in  $lI_d$ . Applying this result to the neural dynamics leads to reformulating equation (2) as  $l\mathbf{v} - \mathbf{W} \cdot S(\mathbf{v}) * g_l = \mathbf{u} * g_l$ . We recognize the definition of the variable  $\bar{\mathbf{v}}$  (which was originally defined according to this relation) such that the network's dynamics (2) is equivalent to

$$\bar{\mathbf{v}} = \mathbf{u} * g_l \quad (14)$$

To prove that (5) implements the gradient descent of the relative entropy, we need to use a time-scale separation assumption, enabling the input to reveal its dynamical structure through ergodicity. Indeed, as learning is very slow compared to the activity  $\mathbf{v}$  we can assume that the learning rate  $\epsilon$  is very small compared to 1. In this case, we can apply classical results of periodic averaging [38] to show that the evolution of  $\mathbf{W}$  is well-approximated by

$$\dot{\mathbf{W}} = [\bar{\mathbf{v}} * \Delta_\gamma + l\bar{\mathbf{v}} - \mathbf{W} \cdot S(\bar{\mathbf{v}})] \cdot S(\bar{\mathbf{v}})' \quad (15)$$



where  $\Delta_\gamma : t \mapsto \frac{\gamma}{2}(g_\gamma(-t) - g_\gamma(t))$ . In the equation above,  $\mathbf{v}$  is the solution of equation (2) such that we can use relation (14). Besides, by going into the Fourier domain, we prove (see below) that

$$[\mathbf{x} * \Delta_\gamma] \cdot \mathbf{y}' = (\dot{\mathbf{x}} * g_\gamma) \cdot (\mathbf{y} * g_\gamma)'$$

Therefore, with equation (1), the averaged equation above becomes

$$\dot{\mathbf{W}} = [\xi(\tilde{\mathbf{u}}) * g_\gamma] \cdot [S(\tilde{\mathbf{u}}) * g_\gamma]' + l\tilde{\mathbf{u}} \cdot S(\tilde{\mathbf{u}})' - \mathbf{W} \cdot S(\tilde{\mathbf{u}}) \cdot S(\tilde{\mathbf{u}})'$$

where  $\tilde{\mathbf{u}} = \mathbf{u} * g_l$ . If  $\mathbf{u}$  is slow enough, i.e.  $\mathbf{u} * g_l * g_\gamma \simeq \mathbf{u}$ , then this equation is precisely the gradient descent of  $H_{\mathbf{u}}$ . If  $\mathbf{u}$  is too fast, then the network will only learn and replay the slow variations of the inputs. Some fast-varying information is lost in the averaging process, mainly because (2) acts as a relaxation equation which filters the activity. Note that this loss of information is not necessarily a problem for the brain, since it may be extracting and treating information at different time scales [39]. Actually, the choice of the parameter  $l$  specifies the time scale under which the inputs are observed.

Stability, convergence to a fixed point of the learning rule (5) follows immediately by applying Krasovskii-Lasalle invariance principle [40].

## STDP as a differential operator

We justify here the following equality  $[\mathbf{x} * \Delta_\gamma] \cdot \mathbf{y}' = (\dot{\mathbf{x}} * g_\gamma) \cdot (\mathbf{y} * g_\gamma)'$ . This is the key mathematical mechanism that makes STDP a good approximation of the temporal derivative of the inputs.

We proceed in two steps, both of which consist in going in the Fourier domain, where convolutions are turned into multiplications and reciprocally. We use the unitary, ordinary frequency convention for the Fourier transform. Observe that the Fourier transform of  $g_\gamma$  is  $\hat{g}_\gamma(\xi) = \frac{\gamma}{\gamma + 2i\pi\xi}$ . And we define  $g'_\gamma : t \mapsto g_\gamma(-t)$  such that  $\Delta_\gamma = \frac{\gamma}{2}(g'_\gamma - g_\gamma)$  and  $\hat{g}'_\gamma(\xi) = \hat{g}_\gamma(-\xi)$ .

1. Let us show that  $\mathbf{x} * \Delta_\gamma = \dot{\mathbf{x}} * \frac{g_\gamma + g'_\gamma}{2}$ .

The Fourier transform of the convolution  $\mathbf{x} * \Delta_\gamma$  is the product  $\hat{\mathbf{x}} \hat{\Delta}_\gamma$ . Besides,

$$\begin{aligned} \frac{2}{\gamma} \hat{\Delta}_\gamma &= \widehat{g'_\gamma - g_\gamma}(\xi) = \frac{\gamma}{\gamma - 2i\pi\xi} - \frac{\gamma}{\gamma + 2i\pi\xi} = 2i\pi\xi \frac{2\gamma}{\gamma^2 + 4\pi^2\xi^2} \\ &= \frac{2i\pi\xi}{\gamma} \left( \frac{\gamma}{\gamma + 2i\pi\xi} + \frac{\gamma}{\gamma - 2i\pi\xi} \right) = \frac{2i\pi\xi}{\gamma} \widehat{(g'_\gamma + g_\gamma)}(\xi) \end{aligned}$$

Because  $\frac{d\hat{\mathbf{x}}}{dt}(\xi) = \hat{\mathbf{x}} 2i\pi\xi$ , taking the inverse Fourier transform of  $\hat{\mathbf{x}} 2i\pi\xi \frac{\widehat{g'_\gamma + g_\gamma}}{2}(\xi)$  gives the result.

2. Let us show that  $[\mathbf{x} * \frac{g_\gamma + g'_\gamma}{2}] \cdot \mathbf{y}' = (\mathbf{x} * g_\gamma) \cdot (\mathbf{y} * g_\gamma)'$

We proceed again in two steps:

- (a) Let us show that  $\frac{g_\gamma + g'_\gamma}{2} = g_\gamma * g'_\gamma$

The Fourier transform of the convolution  $g_\gamma * g'_\gamma$  is the product  $\frac{\gamma}{\gamma + 2i\pi\xi} \frac{\gamma}{\gamma - 2i\pi\xi} = \frac{\gamma^2}{\gamma^2 + 4\pi^2\xi^2}$ . From the usual Fourier tables we observe that the right hand side is the Fourier transform of  $t \mapsto \frac{\gamma}{2} e^{-\gamma|t|}$ . This immediately leads to the result.

- (b) Let us show that  $(\mathbf{x} * g'_\gamma) \cdot \mathbf{y}' = \mathbf{x} \cdot (\mathbf{y} * g_\gamma)'$

Compute

$$\begin{aligned} \{(\mathbf{x} * g'_\gamma) \cdot \mathbf{y}'\}_{ij} &= \int_{-\infty}^{\infty} (\mathbf{x}_i * g'_\gamma)(s) \mathbf{y}_j(s) ds = \int_{-\infty}^{\infty} \int_{-\infty}^{+\infty} \mathbf{x}_i(t) g'_\gamma(s - t) \mathbf{y}_j(s) ds dt \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{+\infty} \mathbf{x}_i(t) g_\gamma(t - s) \mathbf{y}_j(s) ds dt = \{\mathbf{x} \cdot (\mathbf{y} * g_\gamma)'\}_{ij} \end{aligned}$$

Using the result (a) and applying result (b) to  $\mathbf{x} * g_\gamma$  and  $\mathbf{y}$  leads to the result  $[\mathbf{x} * \frac{g_\gamma + g'_\gamma}{2}] \cdot \mathbf{y}' = (\mathbf{x} * g_\gamma) \cdot (\mathbf{y} * g_\gamma)'$ .

Using the first result and applying the second to  $\dot{\mathbf{x}}$  and  $\mathbf{y}$  leads to the result  $[\mathbf{x} * \Delta_\gamma] \cdot \mathbf{y}' = (\dot{\mathbf{x}} * g_\gamma) \cdot (\mathbf{y} * g_\gamma)'$ .

## Acknowledgments

The authors thank Jonathan Touboul and Olivier Faugeras for their helpful comments and suggestions.

MNG was partially funded by the Amarsi project (FP7-ICT #24833), the ERC advanced grant NerVi #227747, the région PACA, France and the IP project BrainScaleS #269921.

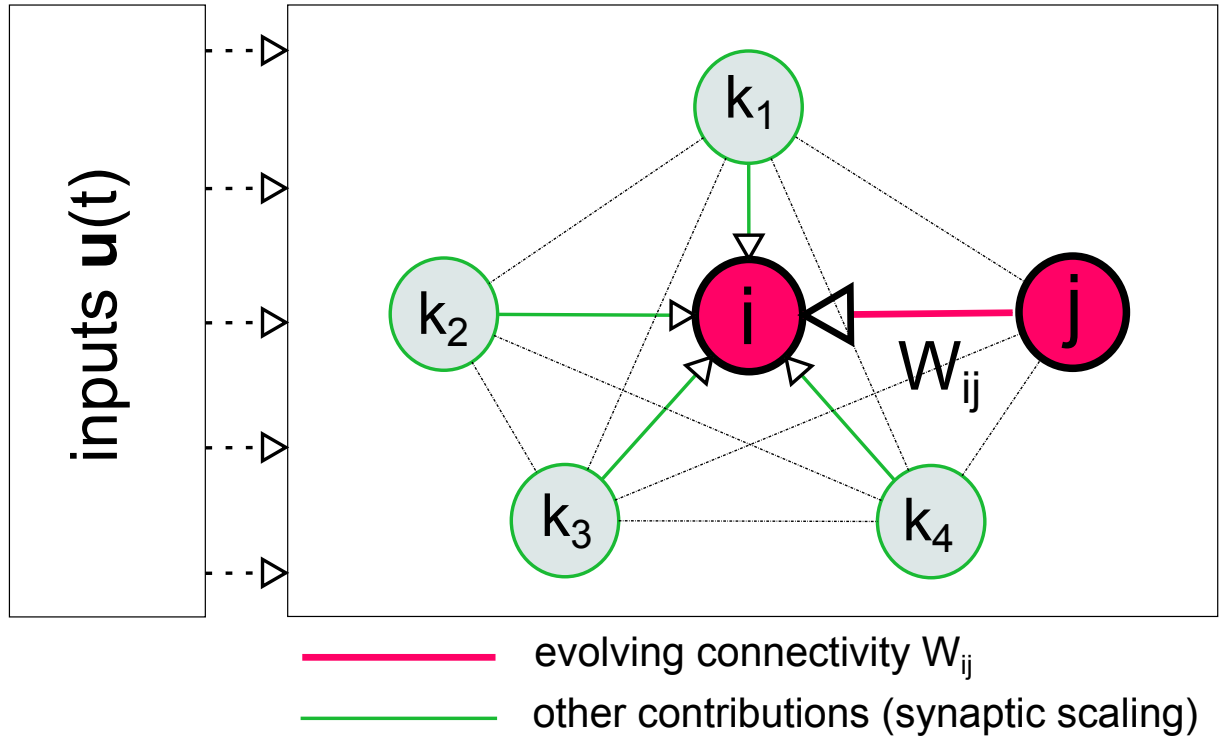
## References

1. Hebb D (1949) The organization of behavior: A neuropsychological theory.
2. Markram H, Lübke J, Frotscher M, Sakmann B (1997) Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps. *Science* 275: 213–215.
3. Bi G, Poo M (1998) Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *The Journal of Neuroscience* 18: 10464–10472.
4. Abbott L, Nelson S (2000) Synaptic plasticity: taming the beast. *Nature neuroscience* 3: 1178–1183.
5. Turrigiano G, Nelson S (2004) Homeostatic plasticity in the developing nervous system. *Nature Reviews Neuroscience* 5: 97–107.
6. Bienenstock E, Cooper L, Munro P (1982) Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *The Journal of Neuroscience* 2: 32–48.
7. Hennequin G, Gerstner W, Pfister J (2010) Stp in adaptive neurons gives close-to-optimal information transmission. *Frontiers in Computational Neuroscience* 4.
8. Linsker R (1988) Self-organization in a perceptual network. *Computer* 21: 105–117.
9. Olshausen B (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* 381: 607–609.
10. Bell A, Sejnowski T (1997) The independent components of natural scenes are edge filters. *Vision research* 37: 3327–3338.
11. Pearlmutter B (1995) Gradient calculations for dynamic recurrent neural networks: A survey. *Neural Networks, IEEE Transactions on* 6: 1212–1228.
12. Kenet T, Bibitchkov D, Tsodyks M, Grinvald A, Arieli A (2003) Spontaneously emerging cortical representations of visual attributes. *Nature* 425: 954–956.
13. Berkes P, Orbán G, Lengyel M, Fiser J (2011) Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science* 331: 83.
14. Friston K (2010) The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience* 11: 127–138.

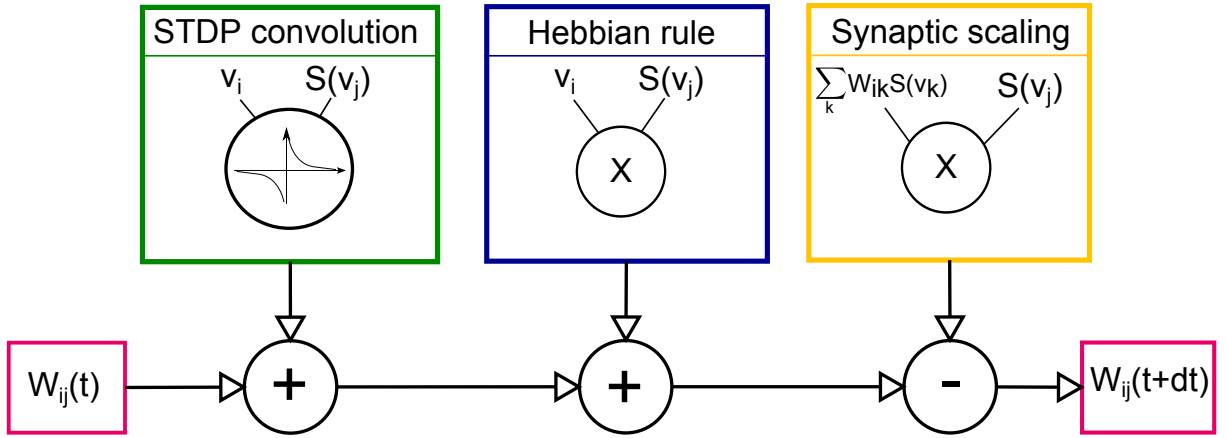
15. Funahashi K, Nakamura Y (1993) Approximation of dynamical systems by continuous time recurrent neural networks. *Neural networks* 6: 801–806.
16. Ackley D, Hinton G, Sejnowski T (1985) A learning algorithm for boltzmann machines. *Cognitive science* 9: 147–169.
17. Hinton GE (2009) Deep belief networks. *Scholarpedia* 4: 5947.
18. Friston K, Harrison L, Penny W (2003) Dynamic causal modelling. *Neuroimage* 19: 1273–1302.
19. Galán R (2008) On how network architecture determines the dominant patterns of spontaneous neural activity. *PLoS One* 3: e2148.
20. Potthast R, beim Graben P (2009) Inverse problems in neural field theory. *SIAM Journal on Applied Dynamical Systems* 8: 1405.
21. Gerstner W, Kistler W (2002) Spiking neuron models: Single neurons, populations, plasticity. Cambridge Univ Pr.
22. Sejnowski T (1977) Statistical constraints on synaptic plasticity. *Journal of theoretical biology* 69: 385.
23. Miller K, MacKay D (1994) The role of constraints in hebbian learning. *Neural Computation* 6: 100–126.
24. Caporale N, Dan Y (2008) Spike timing-dependent plasticity: a hebbian learning rule. *Annu Rev Neurosci* 31: 25–46.
25. Gerstner W, Sjöström (2010) Spike-timing dependent plasticity. *Scholarpedia* 5: 1362.
26. Galtier M (2012) A mathematical approach to unsupervised learning in recurrent neural networks. Ph.D. thesis, Mines Paristech / INRIA.
27. Izhikevich E, Desai N (2003) Relating stdp to bcm. *Neural Computation* 15: 1511–1523.
28. Dayan P, Abbott L, Abbott L (2001) Theoretical neuroscience: Computational and mathematical modeling of neural systems. Taylor & Francis.
29. Gerstner W, Kistler W (2002) Mathematical formulations of hebbian learning. *Biological cybernetics* 87: 404–415.
30. Miller K (1996) Synaptic economics: Competition and cooperation in correlation-based synaptic plasticity. *Neuron* 17: 371–374.
31. Van Rossum M, Bi G, Turrigiano G (2000) Stable hebbian learning from spike timing-dependent plasticity. *The Journal of Neuroscience* 20: 8812–8821.
32. Song S, Miller K, Abbott L (2000) Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature neuroscience* 3: 919–926.
33. Turrigiano G, Leslie K, Desai N, Rutherford L, Nelson S (1998) Activity-dependent scaling of quantal amplitude in neocortical neurons. *NATURE* 391: 893.
34. Kim J, Tsien R, Alger B (2012) An improved test for detecting multiplicative homeostatic synaptic scaling. *PloS one* 7: e37364.

35. Oja E (1982) Simplified neuron model as a principal component analyzer. *Journal of mathematical biology* 15: 267–273.
36. Penland C, Sardeshmukh P (1995) The optimal growth of tropical sea surface temperature anomalies. *Journal of climate* 8: 1999–2024.
37. Yeung M, Tegnér J, Collins J (2002) Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences* 99: 6163.
38. Sanders J, Verhulst F (1985) Averaging methods in nonlinear dynamical systems, volume 59. Springer.
39. Kiebel S, Daunizeau J, Friston K (2008) A hierarchy of time-scales and the brain. *PLoS computational biology* 4: e1000209.
40. Khalil H, Grizzle J (1992) Nonlinear systems. Macmillan Publishing Company New York.

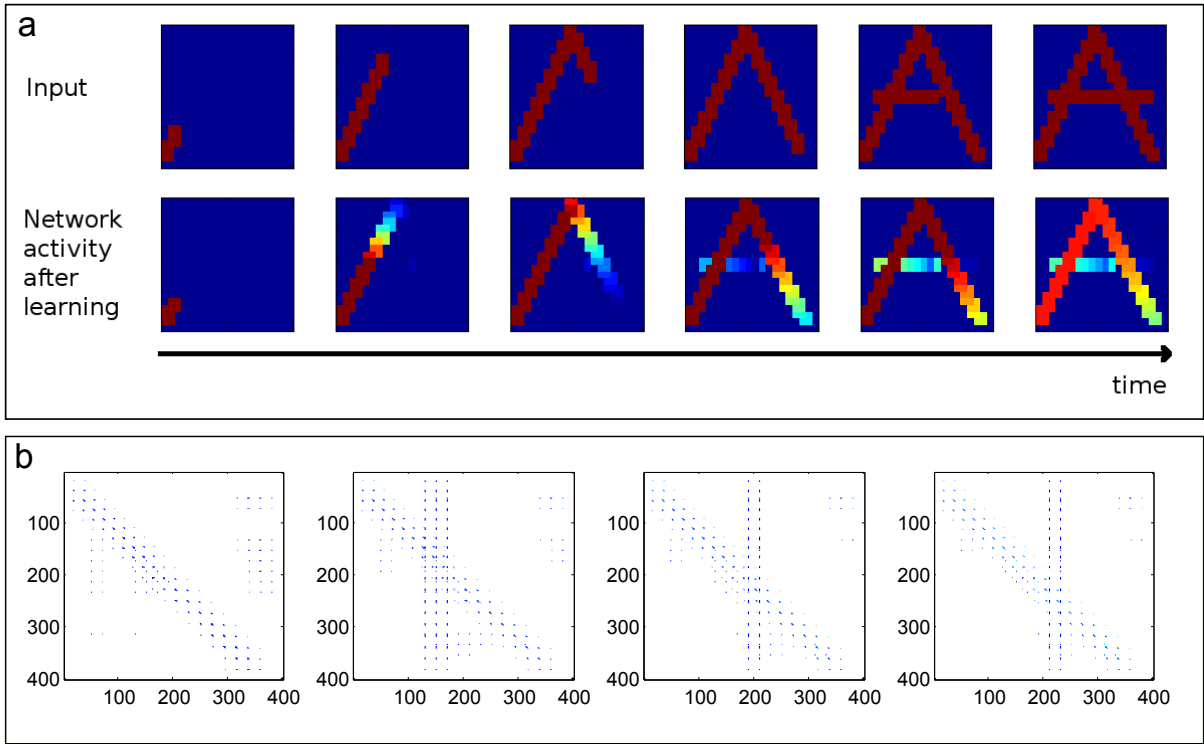
## Figure Legends



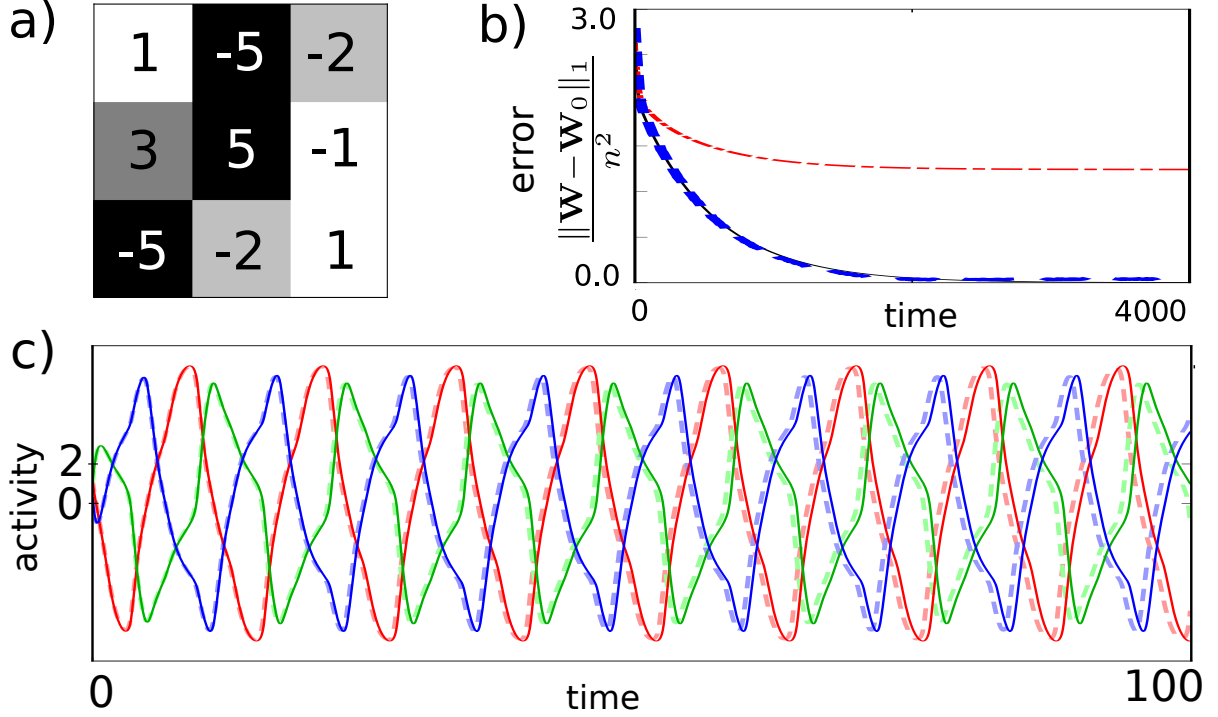
**Figure 1. Schematic picture of the neuronal network.** Each neuron receives an external input and is connected to other neurons in the network. If the connection strength  $W_{ij}$  (pink) from neuron  $j$  to neuron  $i$  evolves according to learning rule (5), then neurons  $i$  and  $j$  contributes to the STDP convolution and Hebbian learning parts, while other neurons  $k_l$  contributes to the synaptic scaling through the synaptic drive received by neuron  $i$ , thus locally accessible by neuron  $i$ .



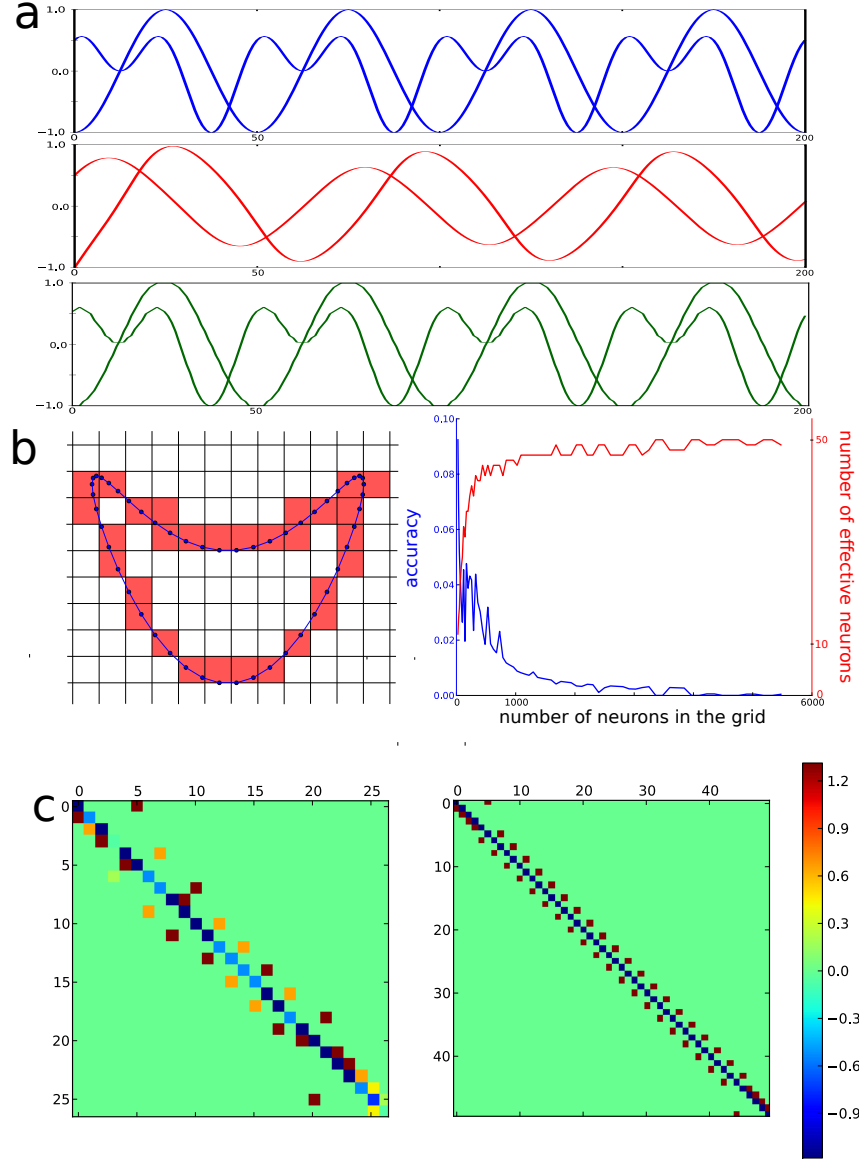
**Figure 2. Schematic representation of the learning rule (5).** The connection strength  $W_{ij}$  between neuron  $j$  and neuron  $i$  is dynamically modified according to the right-hand side of (5). The first term can be seen as a convolution with a SDTP-like filter (green box). The second term appears in the form of a product between the activity of neuron  $i$  and the post-synaptic activity of neuron  $j$ , therefore called Hebbian rule (blue box). Eventually, the last term involves a product between the synaptic drive received by neuron  $i$  and the post-synaptic activity of neuron  $j$ . Modifying  $W_{ij}$  with a negative sign, this term may account for synaptic scaling mechanisms. See the **Discussion** for a detailed biological interpretation of each term.



**Figure 3. Learning how to write the letter A.** a) Time evolution of the input movie (top row) and of the network activity after learning (bottom row). Each pixel corresponds to a neuron. b) Evolution of the connectivity matrix at different times during the learning phase. Parameters: Number of neurons  $n = 400$ ;  $l = 1$ ;  $S(x) = \tanh(x)$ .



**Figure 4. Retrieving the connectivity** a) Connectivity matrix  $\mathbf{W}_0$  of the input network. b) Evolution of the difference between current connectivity and input connectivity through learning. The red dot-dashed black curve corresponds to the online learning rule (5) in the homogeneous case, i.e.  $l = l_{net} = l_0$  in (5) and (2). The blue dashed curve corresponds to the online learning rule (5) in the hybrid framework, i.e.  $l = l_0$  in (5) and  $l = l_{net} \gg l_0$  in (2). For this simulation we chose  $l_{net} = 50$ . The black curve corresponds to the batch relative entropy minimization (4). c) The dashed curves correspond to the activity of the inputs. It is a three dimensional input and the three different colors correspond to the different dimensions. The plain curves correspond to the simulation of network (2) post learning, in the hybrid framework, and without inputs. The parameters for these simulations are  $l_0 = 1$ ,  $\epsilon = 0.01$  and  $\gamma = 100$ .



**Figure 5. Learning the low dimensional inputs with high dimensional networks** a) (top row) The inputs are displayed in blue. (middle row) The naive application of the learning rule (4) to a two dimensional network leads to the approximation in red. (bottom row) The reconstruction of the inputs for a network copying the dynamics in the phase plane. This corresponds to plotting the different components of the observed trajectory in the phase plane which is created autonomously by the network post-learning. We used  $n = 3364$  neurons in this figure. b) (left) Discretization of the phase space of the inputs. The inputs are plotted in blue in their phase space. Only the neurons corresponding to the red boxes are active during learning. (right) In blue, the accuracy of the approximation post-learning for different number of neurons in the grid above. In red, the number of neurons really involved in learning  $n_{eff}$  corresponds to the number of red boxes in the left picture of figure b. c) Equilibrium connectivity for different number of neurons in the network:  $n = 10^2 = 100$  neurons for the left picture and  $n = 58^2 = 3364$  for the right picture. The irregularity (non convolutionality) of the connectivity is due to the fact a circle is irregularly projected on the rectangular grid. The parameters used in this simulation are  $l = 10$ , for figure a and  $l = 0.1$  for figures b and c,  $S(x) = \tanh(x)$  and the input is temporally discretized on  $T = 50$  points.